Module: Core Java

Session 11: Abstract Classes and Interfaces Practice

* This is a practice session; you will work on Abstract Classes and Interfaces assignments.
* You can discuss your doubts with the trainer

**Assignments:**

**Assignment 1 – Abstract class assignment**

Create a class Student.

Instance Variables: name, class

Methods: abstract: getPercentage( ), static getTotalNoStudents( )

Constructors: Initialize name and class

b). Create a class ScienceStudent that inherits from Student

Instance variables: phisicsMarks, chemistryMarks, mathsMarks

Class variable:noOfStudents

getPercentage method: computes and returns the percentage of marks(Max marks for a subject :100

c). Create a class HistoryStudent that inherits from Student

Instance variables: historyMarks, civicsMarks (Max Marks for a subject : 100),

Class variables: noOfStudents

getPercentage method: computes and returns the percentage of marks(Max marks for a subject :100

d). Create a general class AllStudents. In this class create some history, science students. Assign marks. Show the percentage of marks for each student. Also find the total no students.

**Assignment 1 – Abstract class assignment**

Create a Shape as an abstract class having the methods draw and area methods , Implement the classes for different shapes like Circle, Rectangle, Square in such way that , based on the shape the areas of the shapes to be calculated and displayed.

Create abstract class and its implemented classes are as follows:

AbstractClassName : Shape

Abstract methods: draw() :void

area() : double

display():void

Implemented classes are as follows:

ClassNames : Rectangle , Circle and Square implementing the all abstract methods with their own implementationrs.

Class name:Rectangle

Instance variables: length : double , breadth :double

Class name: Circle

Instance variables: radius : double

Class name: Square

Instance variables: side : double

**Assignment 3 – Interface Assignment**

Create the classes and interfaces as following.

**Interfaces:**

1. InterfaceOne

methodOne()

2. InterfaceTwo

methodTwo()

3. InterfaceThree

methodThree()

4. InterfaceFour

methodFour()

**Classes:**

1. abstract ClassOne ( implements InterfaceThree)

a. Implement methodOne, methodTwo

2. ClassTwo (extends ClassOne implements InterfaceFour)

a. Implements methodThree, methodFour

Write a drive program and create object for classTwo and call all the methods. Create reference for classOne and class methods on classOne.

**Assignment 4 – What will be output of following program**

1. interface Apple{

float cost=9.5f;

public void display();

}

class Try implements Apple{

public static void main(String[] args){

Try t=new Try();

t.display();

}

public void display(){

System.out.print("cost of apple is :"+cost);

}

}